

Opinnäytetyö (AMK)
Tietotekniikka
Hyvinvointiteknologia
2015

Arttu Nurminen

TERVEYSTIETOKANTOJEN KEHITTÄMINEN WWW- YMPÄRISTÖSSÄ



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

Arttu Nurminen

TERVEYSTIETOKANTOJEN KEHITTÄMINEN WWW-YMPÄRISTÖSSÄ

Tämän opinnäytetyön tarkoituksena oli tutkia terveystietojen tallentamista sähköisiin tietokantoihin. Tietokantojen käyttämistä tarkasteltiin terveysteknologian näkökulmasta. Toimeksiantajalle kehitettiin prototyyppi tietokantaa hyödyntävästä WWW-sovelluksesta, jolla voidaan seurata kuntoutettavan ihmisen kehitystä.

Opinnäytetyön aluksi kerrottiin, miten sähköisten tietokantojen käyttäminen tehostaa ja nopeuttaa terveydenhuollon prosesseja. Tämän jälkeen valaistiin terveysteknologian ja etenkin terveystietokantojen tuotekehitykseen liittyviä erityispiirteitä, joita ovat esimerkiksi tiukka lainsäädäntö ja normaalia pidempi tuotekehitysprosessi. Projektiosuudessa tehdyn WWW-sovelluksen taustatiedoiksi kartoitettiin suomalaisia kansallisia terveystietokantapalveluja. Tarkasteltuja palveluja olivat Omakanta ja Taltioni. Näiden avulla voidaan katsoa omia terveystietoja Internet-yhteyden avulla.

Toimeksiantajalle tehty projektityö oli osa laajempaa kuntoutusjärjestelmää. Toimeksiannossa suunniteltiin ja toteutettiin prototyypit tietokannasta ja WWW-sovelluksesta. WWW-sovelluksella voitiin tehdä tietokantahakuja halutun kuntoutettavan henkilön tiedoista. Yksilöllisessä henkilönäkymässä näkyi tarkat tiedot henkilöstä ja kuntoutuksesta kerätyt tiedot. Kuntoutustietojen esittäminen toteutettiin sekä tekstinä että graafisessa muodossa.

Opinnäytetyön tuloksena saatiin kehitettyä vaatimusmäärittelyn mukainen prototyyppi tietokantaa hyödyntävästä WWW-sovelluksesta. Tehty prototyyppi toimii mallina toimeksiantajalle suurempaa kokonaisuutta varten. Opinnäytetyön raportti toimii dokumentaationa lopullisen järjestelmän tuotekehityksessä.

ASIASANAT:

Potilastietojärjestelmät, relaatiotietokannat, terveyden edistäminen, verkko-ohjelmointi

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information technology | Wellness technology

2015 | 43

Teppo Saarenpää

Arttu Nurminen

HEALTH DATABASE DEVELOPMENT IN A WWW ENVIRONMENT

The goal of this thesis was to research the storing of health information to electrical databases. The use of databases was examined from a health technology point of view. For the commissioner of this thesis, the purpose of this thesis was to develop a prototype of a database WWW application which can monitor the progress of a person rehabilitating.

The theoretical part of the thesis discusses how the use of electrical databases enhances and expedites the processes of healthcare. After that, the thesis elaborates on health technology and especially the product development of health databases related to characteristics which are, for example, strict legislation and longer than normal product development process. The Finnish national health databases were examined as the background of the WWW application in this thesis project. The examined services were Omakanta and Taltioni. With these, a patient or a user can observe their own health information with an Internet connection.

The project work that was carried out for the commissioner was a part of a larger rehabilitation system. The purpose of this thesis was to plan and implement prototypes of the database and the www application. With the WWW application, the user can carry out database searches about the information related to a specific person in rehabilitation. In the individual person view, the application displays accurate information of the person and the information gathered from rehabilitation. The presentation of rehabilitation information was implemented both as a text and as graph.

The results of the thesis were the requirement specifications according to the prototype of a database www application. The resulting application works as a model for the commissioner for a larger entity. The report of the thesis works as a documentation for a product development of the final system.

KEYWORDS:

Patient information systems, relational databases, health promotion, network programming

SISÄLTÖ

SANASTO	6
1 JOHDANTO	7
2 TIETOKANNAT TERVEYTEKNOLOGIASSA	9
2.1 Terveysteknologia	9
2.2 Terveystietokannat	12
3 KANSALLISET TERVEYSTIETOKANNAT	15
3.1 Omakanta	15
3.2 Taltioni	17
4 TIETOKANTAA HYÖDYNTÄVÄN WWW-SOVELLUKSEN KEHITTÄMINEN	19
4.1 Kehitysvaiheet	19
4.1.1 Tietokanta	19
4.1.2 WWW-sovellus	22
4.2 Kehitystyökalut	23
4.2.1 Microsoft Visual Studio 2010	23
4.2.2 Microsoft SQL Server 2008	25
4.3 Kehitysmenetelmät	26
4.3.1 .NET-ohjelmistokomponenttikirjasto	26
4.3.2 ASP.NET-WWW-ohjelmistokehys	27
4.4 Kielet	29
4.4.1 Visual C#	29
4.4.2 XHTML	31
4.4.3 CSS	32
4.4.4 JavaScript	33
4.4.5 SQL	34
4.5 Toteutus	35
4.5.1 Tietokannan luominen	36
4.5.2 WWW-sovelluksen kehittäminen	38
5 YHTEENVETO	40
LÄHTEET	42

KUVAT

Kuva 1. Esimerkki tietokannan taulujen välisestä suhteesta.	22
Kuva 2. Tietokannan luonti Microsoft Visual Studio graafisella työkalulla.	37
Kuva 3. Tietokannan taulujen relaatiot.	38
Kuva 4. Näkymä jalan kuntoutustiedoista WWW-sovelluksessa.	39

TAULUKOT

Taulukko 1. Esimerkki tietokannan taulusta.	20
---	----

SANASTO

ANSI	American National Standards Institute. Standardien kehittämistä Yhdysvalloissa valvova järjestö.
ASP	Active Server Pages. Microsoftin luoma WWW-palvelinympäristön ohjelmointiin suunniteltu menetelmä.
CE-merkintä	Conformité Européenne. Merkinnällä valmistaja lupaa, että tuote on tarvittavien EU-direktiivien mukainen ja on läpäissyt määritetyt tarkistukset.
CSS	Cascading Style Sheets. Tyyliohjekieli, jolla määritellään käytettävät tyyliohjeet samaan paikkaan.
ECMA-järjestö	European Computer Manufacturers Association. Kansainvälisiä tietotekniikka-alan standardeja määrittelevä standardointijärjestö.
HTML	Hypertext Markup Language. WWW-sivujen merkintään kehitetty merkintäkieli.
HTTP	Hypertext Transfer Protocol. Tiedonsiirtoon suunniteltu Internet-protokolla.
IBM	International Business Machines. Tietokoneiden ja palvelimien valmistaja.
IEC	International Electrotechnical Commission. Kansainvälisiä sähköalan standardeja määrittelevä standardointijärjestö.
ISO	International Organization for Standardization. Kansainvälisiä standardeja määrittelevä standardointijärjestö.
MD-direktiivi	Directive on Machinery. Lääkinnällisten laitteiden EU-direktiivi.
PHP	Hypertext Preprocessor. PHP Groupin kehittämä WWW-palvelinympäristöön suunniteltu ohjelmointikieli.
SIM-kortti	Subscriber Identity Module. Matkapuhelinliittymän tilaajan yksilöivä kortti, joka sisältää sähköisen identiteettiavaimen.
XHTML	Extensible Hypertext Markup Language. WWW-sivujen merkintään kehitetty merkintäkieli, joka on yhteensopiva XML:n muutosääntöjen kanssa.
XML	Extensible Markup Language. Tiedon merkityksen kuvaukseen kehitetty merkintäkieli.
WWW	World Wide Web. Hypertekstijärjestelmä, jota käytetään Internet-yhteyden avulla.

1 JOHDANTO

Ihmisten terveystietoja on alettu tallentamaan yhä enemmän sähköiseen muotoon, ja näiden tietojen tallentaminen alkaa jo syntymästä. Tietoja kerääntyy todella paljon ajan kuluessa. Omien terveystietojen hallinta paperisessa muodossa voi sekin jo tuottaa haasteita, puhumattakaan esimerkiksi jonkin ison yhteisön kaikkien ihmisten terveystiedoista. Ison tietomäärän varastointi ja hakeminen onnistuu tehokkaimmin sähköisillä apuvälineillä. Näistä apuvälineistä käytetyin on sähköinen tietokanta. Terveystietoja sisältäviä tietokantoja kutsutaan terveystietokannoiksi.

Omat terveystiedot eivät ole välttämättä aina tallella. Esimerkiksi rokotuskortti voi olla hukassa ja edellisistä rokotuksista voi olla niin paljon aikaa, ettei muista, koska täytyy seuraavaksi ottaa säännöllisin väliajoin otettava rokotus. Omia terveystietoja on aiemmin ollut mahdollista kysellä joko käymällä terveydenhuollon toimipisteessä tai ottamalla sinne yhteyttä puhelimitse. Tällä tavalla terveydenhuollon toimipisteen henkilökuntaan kuuluva on voinut kertoa kysytyjä tietoja paikallisesti tietokoneelta, jolla on verkkoyhteys lähiverkossa sijaitsevaan tietokantapalvelimeen. Tämä on usein vaatinut jokaiseen päätietokoneeseen asennetun erillisen asiakasohjelman, johon kirjaudutaan ja haetaan tiedot palvelimelta. Mainittu menetelmä vie paljon työaikaa terveydenhuollon henkilökunnalta. Omien terveystietojen selvittäminen on todennäköisesti hankalampaa tällä tavalla, koska puhelinpalvelussa voi olla ruuhkaa tai toimipisteeseen on vaikeaa mennä. Ratkaisuna tähän haasteeseen on kehitetty WWW-selaimen kautta toimivat terveystietokantapalvelut. Näiden palveluiden avulla voidaan tarkastella Internet-yhteyden avulla omia terveystietoja paikasta tai ajasta riippumatta.

Tämän opinnäytetyön tarkoituksena oli tutkia erilaisia Suomessa käytössä olevia kansallisia terveystietokantapalveluja sekä kehittää toimeksiantajalle tietokantaa hyödyntävä WWW-sovellus. Tutkittuja kansallisia palveluja olivat Oma-kanta ja Taltioni. Palvelut ovat tunnettuja Suomessa. Toimeksiantajalle kehitetyn WWW-sovelluksen tarkoituksena oli hakea tallennettuja tietoja tietokannasta

ja näyttää niitä WWW-selaimessa halutussa muodossa, joita olivat tekstimuoto ja graafiset kaaviot. WWW-sovellus oli osa laajempaa kuntoutusjärjestelmää, jota toimeksiantaja oli kehittämässä. WWW-sovelluksen kehittäminen on jaettu kehitysvaiheisiin, kehitystyökaluihin, kehitysmenetelmiin, kieliin ja toteutukseen.

2 TIETOKANNAT TERVEYTEKNOLOGIASSA

Tekniikka on kehittynyt huimasti viime vuosikymmeninä. Tietokoneet ovat helpottaneet ihmisten elämää monissa eri asioissa. Automatiikka on pelastanut ihmisiä loukkaantumisilta ja kuolemilta. Esimerkiksi autoihin kehitetyt automaattisesti laukeavat turvatyynyt lieventävät onnettomuuksien seurauksia.

Terveydenhuoltoalalle on myös kehitetty terveyttä edistäviä laitteita ja sovelluksia. Näillä pyritään ratkaisemaan terveydenhoidossa olevia ongelmia ja haasteita. Laitteet ja sovellukset jaetaan erilaisiin käyttökohteisiin. Terveyttä edistäviä laitteita ja sovelluksia käytetään esimerkiksi terveydenhuoltoalan toimipisteissä ja potilaan omahoidossa. Laitteista ja sovelluksista saatava tieto on hyvä tallentaa järkeviksi kokonaisuuksiksi. Terveyttä koskevat tiedot tallennetaan usein omaksi kokoelmaksi, jota kutsutaan terveystietokannaksi.

2.1 Terveysteknologia

Terveysteknologialla pyritään parantamaan ihmisten elämänlaatua. Terveyden edistäminen on yksi tärkeimmistä asioista, sillä se vaikuttaa merkittävästi ihmisten elämänlaatuun. Tämän vuoksi terveysteknologia pureutuu vahvasti terveydenhuollon toimialalle. Terveysteknologia kehittää monenlaisia terveyden edistämistä tukevia laitteita ja sovelluksia. Terveydenhuoltoalan toimipisteissä olevia laitteita ovat esimerkiksi automaattinen sydäntoimintojen mittaaaja tai defibrillaattori. Terveydenhuoltoalalla olevia sovelluksia ovat esimerkiksi potilastietojärjestelmät. Potilastietojärjestelmät perustuvat terveystietojen nopeaan sähköiseen varastointiin ja hakemiseen. Monien sairauksien omahoitoon on kehitetty erilaisia laitteita. Näitä laitteita on esimerkiksi verensokerin mittaukseen tarkoitettu laite sokeriaineenvaihdunta-sairautta eli diabetesta sairastaville ihmisille. Erityisesti vanhemmille ihmisille on kehitetty monenlaisia elämänlaatua parantavia laitteita. Tällaisia ovat esimerkiksi turvarannekkeet, joilla voidaan hälyttää apua, jos on esimerkiksi kaatunut kotona eikä pysty itse liikkumaan. [1,2]

Terveysteknologia on alana kasvava. Tähän on monia syitä mutta oleellisin on se, että ihmisten terveyttä halutaan edistää entistä enemmän tekniikan avulla. Tämä luo kasvavaa kysyntää terveysteknologisille tuotteille. Lääketieteen kehityksessä myös uusien terveysteknologisten tuotteiden kehitysmahdollisuus kasvaa. [3]

Terveysteknologia erottuu muista teknologian osa-alueista tiukan lainsäädännön vuoksi. Tämä selittyy sillä, että terveysteknologian tuote on tekemisissä ihmisen hyvinvoinnin kanssa, joten joissakin tuotteissa voi olla suurempi riski tuottaa vahinkoa ihmisen terveydelle virhetilanteessa. Suomessa on terveysteknologisille tuotteille oma lakinsa, jonka nimi on ”Laki terveydenhuollon laitteista ja tarvikkeista”. Tuotaessa markkinoille laissa määritellyn lääkinnällisen laitteen tai tarvikkeen, se täytyy hyväksyttää viranomaisella. Tuotteen täytyy olla turvallinen, käyttötarkoitukseen sopiva ja suorituskykyinen. Siinä täytyy olla myös näkyvillä CE-merkintä. [3]

Lääkinnällisen laitteen tai tarvikkeen hyväksymisprosessi vaihtelee, riippuen tuotteen riskiluokituksesta. Prosessissa jaotellaan laite tai tarvike MD-direktiivin mukaan neljään eri riskiluokkaan. Nämä riskiluokat ovat 1, 2a, 2b ja 3. Riskiluokassa 1, tuotteella on mahdollisuus haavoittaa vähän ihmiskehoa jos laite menee epäkuntoon tai rikki. Riskiluokassa 3 tuotteella on mahdollisuus haavoittaa paljon ihmiskehoa. Riskienhallintaan liittyvät standardit auttavat riskien yhtenäisessä määrittelemisessä. [3]

Terveydenhuoltoala ja sen markkinat ovat tiukasti säädelty kohdemaan lainsäädännöstä riippuen. Esimerkiksi Yhdysvalloilla ja Kanadalla on molemmilla oma lainsäädäntönsä terveysteknologisten tuotteiden hyväksymisestä mutta esimerkiksi Euroopan unionissa on yhteinen koko unionin kattava hyväksymisdirektiivi. Eri kohdemarkkinoiden lainsäädäntö eroaa toisistaan markkinoille tuotavien tuotteiden rekisteröinnissä ja valvonnassa. [3]

Tuotaessa Suomen markkinoille lääkinnällisen laitteen tai tarvikkeen, se täytyy hyväksyttää Suomen valvovalla viranomaisella Valviralla. Tuote täytyy auditoida puolueettomalla kolmannella osapuolella, jos tuotteen riskiluokitus on 2a, 2b tai

3. Puolueettomat kolmannet osapuolet ovat Valviran ilmoittamia laitoksia. Laitokset suorittavat tuotteelle klinisiä testejä. Ainoastaan riskiluokassa 1 oleva tuote voidaan päästää suoraan markkinoille, jos valmistaja vakuuttaa Valviralle, että tuote täyttää olennaiset vaatimukset. Terveysteknologiassa on erityisen tärkeää noudattaa oleellisia vaatimuksia ja standardeja. Lääkinnällisen laitteen tuotekehityksessä hyödyllisiä standardeja ovat esimerkiksi laatujärjestelmästandardi ISO 13485, ohjelmistokehityksen elinkaarimallistandardi IEC 62304, riskienhallintastandardi ISO 14971 ja käytettävyyssstandardi IEC 62366. [3]

Terveysteknologisten tuotteiden tuotekehitysprosessi on yleensä normaalia pidempi. Lainsäädännön noudattaminen vie tästä ajasta merkittävän osan. Terveysteknologian alalla ovat pärjänneet parhaiten suuret yritykset, joilla on resursseja ja rahaa pitkiin tuotekehitysprosesseihin. Isojen yritysten kilpailijoiksi ovat tulleet entistä enemmän pienemmät yritykset. Isot yritykset ovat yleensä hitaita mukautumaan uusiin tuotekehitysmuutoksiin raskaan byrokratian takia. Pienten yritysten etuna on usein nopea mukautuvuus tuotekehityksessä ilmenviin muutoksiin. Tuotteiden tuotekehitys vaatii paljon resursseja, sillä terveysteknologiassa ei usein sallita yhtään virhettä. Jokaisesta tuotteen käytön aikana tapahtuneesta virheestä tai vaaratilanteesta on velvollisuus ilmoittaa Valviralle. Alalla vaaditaan monialaista tuntemusta, joista pääalat ovat teknologia ja terveydenhuolto. Alan tiukka lainsäädäntö työllistää myös oikeustieteiden asiantuntijoita. [4]

Terveysteknologian haasteena on myös ihmisten pelko siitä, että teknologia syrjäyttää esimerkiksi monet hoitoalan työtehtävät, minkä seurauksena sairast ihmiset ovat entistä enemmän automatiikan armoilla. Sähkövirran toimituksen ongelmat ovat myös iso riski, joka pitää huomioida monessa laitteessa ja soveluksessa. Terveydenhuollon toimipisteissä on monia potilaan elintoimintoja ylläpitäviä laitteita, joita on esimerkiksi hengityskone. Elintoimintoja ylläpitävien laitteiden toimintavarmuus täytyy olla täysin luotettava. Laitteiden osien ja mekaniikan täytyy kestää tarpeellista rasitusta. Laitteet eivät saa sammua, jos sähkövirransyöttö loppuu. Sähkövirran loppuessa laitteiden täytyy kytkeytyä samaan aikaan varavirtajärjestelmään. Varavirtajärjestelmään tulee kuulua myös gene-

raattori, jolla saadaan sähkövirtaa pahimmassa tilanteessa, jossa sähkökatkos kestää kauan ja varastoitu varasähkövirta loppuu. [1]

Haasteena on myös joidenkin ihmisten vastarinta uudelle teknologialle. Tällaisesta vastarinnasta heijastuu usein pelko. Etenkin lääkinnällisten laitteiden ja tarvikkeiden kohdalla on tärkeää käyttää paljon resursseja tuotteen hyvän käytettävyyden toteutukselle. Näiden laitteiden ja tarvikkeiden käytössä ei saa tulla arvaamisen mahdollisuutta. Noudattamalla käytettävyyssstandardeja, voi luottaa tuotteen olevan käytettävä. Lääkinnällisten laitteiden ja tarvikkeiden myynnin kannattavuus on myös iso kysymys, sillä lääketiede kehittyy nopeasti ja tuotekehitysprosessit ovat pitkät. Suomessa haasteita luo myös se, että esimerkiksi kunnat ovat ostaneet tietoteknisiä järjestelmiä usein ainoastaan hinnan perusteella, mikä on johtanut siihen, että Suomessa on paljon terveysteknologisia tuotteita, joiden tietoja ei pysty saamaan automaattisesti samaan muotoon. Tämä on johtunut järjestelmätoimittajien haluttomuudesta tehdä yhteistyötä. Järjestelmistä olisi kuitenkin vaikeaa tehdä yhteensopivia, sillä silloin tuotteissa pitäisi käyttää monia täsmälleen samanlaisia standardeja. [5]

2.2 Terveystietokannat

Ihmisen syntyessä Suomessa hänestä aletaan heti tallentamaan tietoja, joista ensimmäiset ovat syntymätietoja. Näitä tietoja ovat esimerkiksi vanhempien nimet, syntymäpaikka ja syntymäaika. Ihmisistä kerätään usein elämän varrella paljon myös terveystietoja. Ennen tietokoneiden keksimistä, ihmisten tietoja kerättiin paperille ja koottiin arkistoihin. Tietojen etsimiseen arkistoista kului paljon aikaa. Suomessa henkilötietojen etsiminen on toteutunut melko hyvin, sillä Suomessa on ollut käytössä sosiaaliturvatunnukset jokaiselle Suomen kansalaiselle 1960-luvulta lähtien. Sosiaaliturvatunnukset ovat uniikkeja tunnuksia, joilla voidaan tunnistaa jokainen Suomen kansalainen vaikka henkilöillä olisi sama nimi ja syntymäaika. [6]

Huolimatta siitä, että Suomessa on ollut käytössä uniikit sosiaaliturvatunnukset jokaiselle kansalaiselle, paperisten tietojen etsiminen arkistoista on silti vaatinut

paljon aikaa. Joissakin tilanteissa tietojen löytyminen nopeasti voi olla elintärkeää. Tällainen tilanne voi ilmetä jos esimerkiksi sairaalaan tuodaan hengenvaarallisesti loukkaantunut potilas, joka ei pysty kertomaan omia terveystietojaan esimerkiksi tajuttomuuden takia eikä omaisia saada kiinni. Mainitussa tilanteessa potilasta on haastavampaa alkaa hoitamaan. Tällaisesta potilaasta olisi hyvä tietää esimerkiksi veriryhmä, mahdollisesti käytettävä lääkitys ja lääkeaineallergiat. [7]

Tietokoneiden yleistyessä, ihmisten tietoja on alettu tallentamaan sähköiseen muotoon. Tiedot tallennetaan yleensä sähköiseen tietokantaan. Tietojen tallentamisessa sähköiseen muotoon on monia etuja, joista tärkein on tarvittavien tietojen nopea hakeminen. Internet-yhteyden yleistyttyä, tietoja on voinut hakea paikasta riippumatta. Sähköiseen muotoon tallennetut tiedot mahtuvat pienempään tilaan kuin paperisessa muodossa olevat. Tiedot tallentuvat usein keskitettyyn sähköiseen tietokantaan. Tietokanta pyörii palvelimella, johon mahtuu paljon tietoja verrattuna palvelimen fyysisiin mittoihin. Terveystietoja sisältäviä tietokantoja kutsutaan terveystietokannoiksi. [8,9]

Terveystietokantoihin tallentuneista tiedoista on suurta hyötyä ihmisen tunnistamisessa. Henkilön käydessä esimerkiksi terveydenhuollon toimipisteessä, hänestä tallennetaan kaikki käyntiin liittyvät terveystiedot sähköiseen tietokantaan. Jos sama henkilö tuodaan esimerkiksi ambulanssilla tajuttomana ja akuuttihoidon tarpeessa toiseen terveydenhuollon toimipisteeseen, josta on pääsy aiemmin tallennettuun tietoon tietokannassa, henkilöstä saadaan haettua aiemmin tallennetut terveyteen liittyvät tiedot nopeasti. Tämä edellyttää ainoastaan sitä, että hoitohenkilökunta tietää henkilön sosiaaliturvatunnuksen, joka on voitu löytää esimerkiksi potilaan taskussa olleesta ajokortista rikoslaissa säädetyn pakkotilan nojalla. Henkilö voidaan tunnistaa hyvällä todennäköisyydellä myös hammaskartan avulla, jos hän on käynyt hammaslääkärissä ja hammaskartta on tallentunut tietokantaan. Suomen hyvän julkisen terveydenhuollon ansiosta melkein kaikista Suomen kansalaisista on tallessa hammaskartta. Hammaskarttaan perustuvaa tunnistusta käytetään esimerkiksi onnettomuustilanteissa, joissa uhria ei pystytä muuten tunnistamaan. [10]

Itsestä kerättyjä tietoja on oikeus saada tarkastella henkilötietolain tarkastusoikeuden perusteella. Tarkastaminen tapahtuu ottamalla yhteyttä tietojen kerääjätahoon. Nykyään pystyy myös tarkastelemaan joitakin omia tietoja lähtemättä kodistaan. Etätarkastelu tapahtuu nettiyhteyden ja päätelaitteen kautta. Etätarkastelun tarjoajan nettisivuilla tunnistaudutaan yleensä verkkopankkitunnusten avulla. Verkosta pystyy tutkimaan esimerkiksi missä kaikkialla on asunut tai millaisia terveystietoja itsestä on tallennettu. Etätarkastelun mahdollisuus vähentää palveluntarjoajan työntekijöiden työmäärää, sillä sosiaalista palvelua ei tarvitse enää tehdä niin paljon. Omista tiedoista kiinnostuneet ihmiset säästävät aikaa etätutkimisella, koska heidän ei tarvitse mennä fyysisesti jonottamaan palveluntarjoajalle. Etätarkastelu helpottaa etenkin liikuntarajoitteisten arkea. [11]

Sähköisissä tietokannoissa, joissa on ihmisten tietoja, on myös haasteita. Sähköiset tietokannat pitää olla hyvin suojattuja. Niihin täytyy olla ulkopuolisilta pääsy estetty. Esimerkiksi potilasasiakirjojen salassapito on säädetty ”Laki potilaan asemasta ja oikeuksista” -laissa. Tietokannoista täytyy olla riittävät varmuuskopiot. Jos tietokantoja sisältävä palvelin hajoaa, tiedot voivat olla lopullisesti menetetty. Kaikkein tärkeimmät tiedot olisi kuitenkin hyvä olla paperisena kopiona. Tietokantapalvelimet sijaitsevat yleensä palvelinhuoneissa. Tämä luo lisäkustannuksia, sillä palvelinhuonetta täytyy esimerkiksi jäähdyttää. Terveystietoja sisältävien tietokantapalvelimien täytyy olla erittäin luotettavia, etenkin jos ne sijaitsevat kriittisessä terveydenhuollon toimipisteessä. [12]

3 KANSALLISET TERVEYSTIETOKANNAT

Sähköisten tietokantojen yleistyessä on alettu kehittää terveystietokantoja, jotka toimivat kansallisesti. Kansallisten terveystietokantojen etuna pidetään sitä, että ne voidaan räätälöidä vain tiettyyn käyttöön, jolloin kielikysymyksien, lainsäädännön, standardien ja direktiivien mukainen toteutus onnistuu ilman mutkia. Seuraavaksi on tarkasteltu kahta kansallista terveystietokantapalvelua. Palveluiden nimet ovat Omakanta ja Taltioni. Palveluja yhdistää se, että käyttäjä pystyy tarkastelemaan itsestä tallennettuja terveystietoja Internet-yhteyden välityksellä.

3.1 Omakanta

Omakanta on suomalainen sähköinen verkkopalvelu. Palvelu on tarkoitettu terveystietojen tarkasteluun. Palvelun käyttö on maksutonta. Se kuuluu Kansalliseen Terveysarkistoon. Kansallista Terveysarkistoa kutsutaan myös Kanta-palveluksi. Tämän palvelun takana ovat Sosiaali- ja terveysministeriö, Terveysten ja hyvinvoinnin laitos sekä Kela. [13]

Omakanta-palvelun ideana oli kehittää terveystietoja sisältävä tietokanta, jonka tietoja pystyisi tarkastelemaan WWW-käyttöliittymän avulla. Palvelusta on mahdollista nähdä itseään koskevat terveystiedot ja määrätyt sähköiset reseptit. Tiedot tallentuvat Potilastiedon arkistoon, joka on Kanta-palvelu. Tietojen tallentamisesta palveluun täytyy aluksi saada lupa kohdehenkilöltä. Tietojen tallennusmahdollisuus on käytössä siitä lähtien, kun asioimiseen käytetty terveyspalvelujen tarjoaja on liittynyt palveluun. Palvelu on käytössä suurimmassa osassa julkisia ja yksityisiä terveydenhuollon toimipisteitä. Suurimpien yritysten pilotoimana monet yksityisistä terveydenhuollon toimipisteistä siirtyivät palveluun vuoden 2014 aikana. Julkinen terveydenhuolto puolestaan aloitti palvelun käyttöönoton jo yksityistä sektoria aiemmin. [13]

Palvelu on käytössä jokaiselle täysi-ikäiselle suomalaiselle, jolla on Suomessa käytettävä sosiaaliturvatunnus. Palveluun tallentuu terveystiedot ja sähköiset

reseptit myös alaikäisistä kansalaisista. Alaikäisinä kirjatut tiedot pystyy katso-
maan vasta, kun kyseinen henkilö on saavuttanut täysi-ikäisyyden. Vanhemmat
eivät pysty tällä hetkellä tarkastelemaan alaikäisten lastensa terveystietoja tai
määrättyjä sähköisiä reseptejä. Vanhemmat pystyvät kuitenkin anomaan raport-
tia, josta ilmenee yhteenvetona heidän alaikäisten lastensa terveystiedot ja heil-
le määrättyt sähköiset reseptit. Anominen tapahtuu apteekissa tai terveyden-
huollon toimipisteessä. [13]

Terveyteen liittyviä tietoja on monipuolisesti tarjolla Omakanta-palvelussa. Siel-
tä voi tarkistaa missä terveydenhuollon toimipisteissä tai apteekeissa on asioi-
nut. Omien tietojen tarkastelun suorittanut toimipiste on selvitettävissä lokitie-
doista tietosuojan vuoksi. Palvelua käyttävältä terveydenhuollon ammattilaiselta
edellytetään henkilökohtaista ammattikorttia, jonka on myöntänyt Väestörekiste-
rikeskus. Terveystietoja säilytetään palvelussa siihen asti kuin laki kussakin ta-
pauksessa edellyttää. [13]

Itselleen määrättyt sähköiset reseptit ja niiden tiedot on mahdollista selvittää
palvelusta. Muilla tavoilla kirjoitetut reseptit eivät tallennu palveluun. Sähköiset
reseptit säilyvät 2,5 vuotta Reseptikeskukseksi kutsutussa paikassa ja tämän
jälkeen ne siirtyvät kymmeneksi vuodeksi Reseptiarkistoon, jossa ne eivät ole
enää terveydenhuollon tai apteekkien käytössä vaan ainoastaan viranomaisten
ja tieteellisten tutkimusten käyttöön. [13]

Omakanta helpottaa myös terveydenhuollon ammattilaisten työntekoa. Anta-
malla luvan omien terveystietojen käyttöön, tiedot ovat nähtävissä missä päin
tahansa Suomea. Tällä tavoin esimerkiksi terveydenhuoltopalvelua käyttävän
ihmisen hoitokäynnistä tulee tehokkaampi. Suostumuksen terveystietojen käyt-
töön voi myös perua ja muuten lupa on voimassa toistaiseksi. [13]

Omakanta-palvelu vaatii Internet-yhteyden. Sitä käytetään WWW-selaimen
kautta. Palvelu on henkilökohtainen ja tästä syystä sinne täytyy kirjautua, jotta
voi tarkastella omia terveyteen liittyviä tietoja. Kirjautuminen tapahtuu palvelun
nettisivulta, jonka osoite on www.kanta.fi/omakanta. Palveluun voi tunnistautua
verkkopankkitunnuksilla, sähköisellä henkilökortilla tai mobiilivarmenteella. [13]

3.2 Taltioni

Taltioni on suomalainen sähköinen terveys- ja hyvinvointipalvelu. Sinne voi tallentaa omia terveystietoja. Tiedot löytyvät aina helposti yhdestä paikasta. Palvelulle riittää kysyntää, sillä sähköisten terveyspalveluiden käyttö on tulossa entistä suositummaksi suomalaisten keskuudessa. Palvelulla on yli 15 000 suomalaista käyttäjää ja määrässä tapahtuu 4 %:n vuotuista kasvua. Palveluun tallennetut tiedot on mahdollista jakaa halutulle terveydenhuollon ammattilaiselle tai toiselle Taltionin käyttäjälle. Palvelussa on tietoturva etusijalla. [14]

Palveluntarjoajana toimii Taltioni-osuuskunta, joka on voittoa tavoittelematon palveluntarjoaja. Osuuskunta on perustettu vuonna 2012. Sen omistavat monet terveys- ja hyvinvointitoimijat, jotka tarjoavat terveyspalveluja yksityisellä ja julkisella puolella. Palvelua kehittää laaja palvelukehittäjäverkosto ja jäsenmäärä kasvaa jatkuvasti. [14]

Taltionin ideana on tarjota sähköisessä muodossa oleva terveysarkisto, jossa voi tarkastella terveystietoja. Tämä antaa varmemman vaihtoehdon helpommin häviävillä tiedoilla, kuten esimerkiksi paperiselle rokotuskortille. Yhdessä paikassa olevista tiedoista on helpompi tarkistaa mieleen tulleita asioita kuin alkaa etsiä papereita, jotka eivät välttämättä ole aina tallessa. Palvelu toimii myös ennaltaehkäisevänä terveydenhoitona. [14]

Palveluun voidaan myös tallentaa omaan hyvinvointiin liittyviä asioita perusterveydenhuollon lisäksi. Perusterveydenhuollon tiedoilla tarkoitetaan esimerkiksi rokotuspäivämääriä, käytettyjä lääkkeitä, lääkeaineallergioita ja lääkärikäyntejä. Oman hyvinvoinnin seurantaan tarkoitetuilla työkaluilla pystyy seuraamaan esimerkiksi kuinka hyvin pitää itsestään huolta. Tällaisia asioita ovat fyysinen kunto, paino, unen määrä, ruokavalio ja päihteiden käyttö. Seuranta tapahtuu käytännössä esimerkiksi pitämällä terveys- ja liikuntapäiväkirjaa. [14]

Terveystiliin voi liittää myös muiden henkilöiden tietoja ja jakaa niitä haluttujen henkilöiden kesken. Omien lasten tiedot pystyy helposti lisäämään palveluun. Tällä tavalla monien lasten terveystietojen, kuten esimerkiksi pituuden ja painon

seuranta helpottuu huomattavasti. Palvelun käytettävyyteen on panostettu paljon, sillä Taltioni palvelee vahvasti myös vanhempia ihmisiä. Vanhempien ihmisten on kätevä seurata palvelusta esimerkiksi oman verenpaineen ja kolesteroliarvojen kehitystä. Liittämällä tiliin Taltionin yhteistyökumppaneiden palveluita omasta tilistä saa entistä monipuolisemman. Haluttaessa voi pyytää palvelua käyttävän terveydenhuollon ammattilaisen tallentamaan itsean liittyvät terveystiedot suoraan Taltioni-tiliin. [14]

Palvelun käyttö aloitetaan luomalla oma Taltioni-terveystili. Tilin luonti tapahtuu Taltionin nettisivuilta. Palveluun tunnistaudutaan verkkopankkitunnuksilla tai matkapuhelimen SIM-kortilla olevalla mobiilivarmenteella. Tämän jälkeen omalle tilille voi alkaa tallentamaan terveystietoja. Palveluun on mahdollista liittää Taltionin yhteistyökumppaneiden palveluita. Tilille kirjautuminen onnistuu Taltionin nettisivujen tai ladattavan mobiilisovelluksen avulla. Mobiilisovelluksen käyttöön täytyy luoda erilliset mobiilitunnukset. Palvelua voi käyttää matkapuhelimella myös WWW-selaimen kautta, jolloin erillistä mobiilisovellusta ei tarvitse. [14]

4 TIETOKANTAA HYÖDYNTÄVÄN WWW-SOVELLUKSEN KEHITTÄMINEN

Opinnäytetyön käytännön osuuden tarkoituksena oli kehittää terveystietoja sisältävä tietokanta, josta pystyi hakemaan ja näyttämään tietoja WWW-selaimen kautta. Tietokannasta haettavien tietojen esittämisessä piti käyttää osittain graafista esittämistapaa, joka tarkoitti esimerkiksi kaavioita. Kehitystyöhön liittyvistä asioista on kerrottu viidessä eri kokonaisuudessa. Kokonaisuudet ovat seuraavat: kehitysvaiheet, kehitystyökalut, kehitysmenetelmät, kielet sekä toteutus.

4.1 Kehitysvaiheet

Tietokantaa hyödyntävän WWW-sovelluksen kehittäminen jaetaan usein kahdeksi palaseksi. Ensin suunnitellaan tietokanta, jonne määritellään kaikki tarvittavat taulut, kentät ja relaatiot. Valmiin tietokannan päälle on helpompi kehittää WWW-sovellus, koska silloin tiedetään tarkkaan, millaisia tietoja tietokannasta joutuu hakemaan ja miten niitä täytyy esittää.

4.1.1 Tietokanta

Tietokanta on kokoelma erilaisia tietoja, jotka liittyvät jollain tavalla toisiinsa. Se voi olla yksinkertaisuudessaan paperilappu, jossa on ihmisten nimiä ja heidän puhelinnumeroitaan. Pienten tietomäärien hallinta onnistuu vielä paperilapuilla, mutta jos halutaan esimerkiksi hakea Suomesta tietyn kansalaisen puhelinnumero ja kaikkien ihmisten puhelinnumerot löytyvät paperilapuilta, oikean puhelinnumeron löytäminen nimen perusteella saattaa kestää huomattavan kauan. Laajojen tietokantojen ylläpitämisessä ja tietojen hakemisessa on siirrytty käyttämään sähköisiä tietokantoja. [15,16]

Tietojen tallentamisessa sähköiseen muotoon voidaan käyttää yksinkertaisimmillaan taulukkolaskentaohjelmaa. Taulukkolaskentaohjelmaan pystyy syöttä-

mään tietoja ja tarkastelemaan niitä. Tiedot tallentuvat yhteen tiedostoon ja tiedoston kopioiminen muualle on helppoa. Tietomäärien kasvaessa suuriksi, tietojen haku taulukkolaskentaohjelmalla käy hankalaksi ja aikaa vieväksi. Etenkin erilaisten yhteenkuuluvien tietojen hakeminen kerralla isosta taulukkolaskentatiedostosta on todella hankalaa. Saman tiedon päivittäminen tai muuttaminen monista eri paikoista voi tarkoittaa isoa työmäärää. Isoja sähköisiä tietokantoja on järkevämpi luoda ja päivittää tietokantaohjelmilla. [15]

Tietokantaohjelmat mahdollistavat monimutkaisten tietovarastojen luonnin, haluttujen tietojen nopean haun ja käyttöoikeuksien jaon. Tietokantaohjelmia kutsutaan usein tietokantojen hallintajärjestelmäksi. Näitä järjestelmiä ovat esimerkiksi Microsoft SQL Server, MySQL, Oracle, IBM DB2 ja PostgreSQL. [16]

Tietokanta muodostuu tauluista, jotka pitävät sisällään yhteenkuuluvaa tietoa (Taulukko 1.). Tauluihin määritellään kuinka monta kenttää tarvitaan ja mitä tietotyyppiä ne ovat. Tämän jälkeen tauluja voidaan alkaa täyttämään tietueilla. Taulut koostuvat sarakkeista ja riveistä. Tietokannasta haetaan ja käsitellään tietoja usein kyselykielellä. Kyselykieli pitää sisällään komentoja, jotka muodostavat parametrien kanssa kyselyn, joka ajetaan tietokantaan. Tietoja voidaan syöttää luotuun tietokantaan kolmella eri tavalla: hierarkkisesti, verkkona tai relaatioina. Relaatiomenetelmä on nykyään eniten käytetty tietokantamenetelmä. Sen suosio perustuu käytön helppouteen, joustavuuteen ja laitteistoriippumattomuuteen. [15]

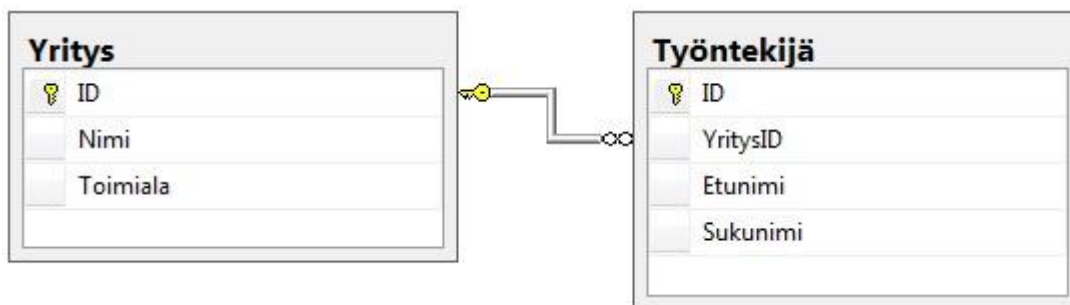
Taulukko 1. Esimerkki tietokannan taulusta.

ID	Etunimi	Sukunimi	Sähköpostiosoite
1	Thomas	Johnson	thomas.johnson@gmail.com
2	Ethan	Smith	ethan.smith@hotmail.com
3	Sarah	Jackson	sarah.jackson@yahoo.com

Relaatiotietokannassa tiedot on jaettu pienemmiksi kokonaisuuksiksi useisiin eri tauluihin ja taulujen välille on määritetty niiden väliset yhteydet. Loogisesti toisiinsa liittyvien tietojen jakaminen eri tauluihin nopeuttaa tietojen päivittämistä ja

hakemista. Nopeus perustuu siihen, että kukin tieto tallennetaan vain yhteen tauluun. Muutettaessa tietyn kentän arvoa, muutosta ei tarvitse tehdä joka paikkaan vaan muutos päivittyy automaattisesti tarvittaviin paikkoihin taulujen välisten yhteyksien ansiosta. Haettaessa vain tiettyjä tietoja tietystä kohteesta, haku voidaan kohdistaa vain niihin tauluihin missä halutut tiedot sijaitsevat. Taulujen välille määritellyt yhteydet voivat olla kolmea eri tyyppiä. Näitä yhteyksiä kutsutaan seuraavilla nimillä: yhdestä yhteen, yhdestä moneen ja monesta moneen. Ensimmäisellä yhteystyypillä tarkoitetaan sitä, että taulun jokaisella tietueella on yhteys vain yhteen toisen taulun tietueeseen. Toisessa yhteystyypissä taulun yhdellä tietueella voi olla yhteys joko yhteen tai useampaan toisen taulun tietueeseen mutta toisinpäin pätee ensimmäisenä mainitun yhteystyyppin sääntö. Viimeisessä yhteystyypissä taulun yhdellä tietueella voi olla yhteys joko yhteen tai useampaan toisen taulun tietueeseen ja sama pätee myös toisinpäin. Viimeistä yhteystyyppiä tulee välttää suunniteltaessa tietokantaa. [17]

Erilaisten yhteyksien luominen taulujen välille tapahtuu avaimien avulla. Avaimella tarkoitetaan taulun tietueen yksilöivää tietoa. Toiseen tauluun loogisesti liittyvän taulun tietueista eli avainehdokkaista valitaan pääavain. Tämän jälkeen toisesta taulusta valitaan pääavaimen tietueeseen liittyvä avain, jota kutsutaan tällöin viiteavaimeksi. Yhdistämällä pääavaimen ja viiteavaimen, taulujen välille syntyy suhde (Kuva 1.). Toimivan ja helposti laajennettavan relaatiotietokannan suunnittelu ei ole kuitenkaan helppoa. Relaatiotietokantaa suunniteltaessa täytyy kiinnittää huomiota todella paljon siihen, ovatko tietyt tietueet liitetty toisiinsa kaikilla mahdollisilla tavoilla, jotka voivat tulla esiin tietoja haettaessa. Relaatiotietokannan laajennettavuus tulee ottaa huomioon jo heti alussa, sillä taulujen ja relaatioiden määrän kasvaessa tietokannan hallinta ja ylläpito vaikeutuvat huomattavasti. [17]



Kuva 1. Esimerkki tietokannan taulujen välisestä suhteesta.

4.1.2 WWW-sovellus

WWW-sovelluksella tarkoitetaan WWW-selaimella käytettävää sovellusta. WWW-selaimia ovat esimerkiksi Firefox, Chrome, Safari, Opera ja Internet Explorer. WWW-selaimen kautta asiakassovellus lähettää tietopyyntöjä tai syöttää tietoja palvelimelle. WWW-sovellus ei tarvitse selaimen lisäksi erillistä asiakasohjelmaa. Useimmat WWW-sovellukset toimivat missä tahansa selaimella ja minkä tahansa päätelaitteen käyttöjärjestelmällä mutta jotkin harvat WWW-sovellukset vaativat tietyn selaimen tai käyttöjärjestelmän toimiakseen oikein. [18]

WWW-sovelluksia on ollut olemassa WWW-selaimien keksimisestä alkaen. Internet-yhteyksien määrän kasvaessa WWW-sovelluksien suosio on kasvanut huomasti, sillä niiden kautta pääsee helposti käsiksi haluttuihin tietoihin ainoastaan pelkän WWW-selaimen kautta. Selaimen kautta toimiva sovellus helpottaa myös sovelluskehitystä, koska sovelluskehittäjät voivat tällöin keskittyä melkein ainoastaan yhden WWW-sovelluksen kehitykseen eikä esimerkiksi eri laitteille tarvitse tehdä omia sovelluksiaan. Laitteistoyhteensopivuus vähentää myös päivityskustannuksia, koska kaikkia sovelluksia ei tarvitse päivittää erikseen. [18]

WWW-sovellus käyttää palvelimella palvelinpuolen ohjelmakoodia. Näitä koodia ovat esimerkiksi ASP ja PHP. Koodit käsittelevät tietojen tallentamista ja tietojen hakua. WWW-palvelimet hallitsevat usein myös tietokantapalvelimella tapahtuvaa toimintaa. [18]

Asiakaspuolella käytettäviä ohjelmakoodoja ovat esimerkiksi HTML ja JavaScript. Koodit huolehtivat WWW-palvelimelta haettujen tietojen esittämisestä käyttäjälle WWW-selaimessa. WWW-selaimessa toimivia asiakassovelluksia ovat esimerkiksi Gmail ja Yahoo –sähköpostiohjelmat. [18]

4.2 Kehitystyökalut

Kehitetyn WWW-sovelluksen kehitystyökaluiksi valikoitui Microsoft Visual Studio 2010 ja Microsoft SQL Server 2008. Valintaan vaikutti eniten halu kokeilla WWW-sovelluksen kehitystä Microsoftin ASP.NET-WWW-ohjelmistokehyksellä, jolloin Visual Studio on luonnollisestiärkevin valinta ja yhteensopivuuden vuoksi tietokantapuoli yhdistettiin SQL Serverin kanssa. Kehitystyökalut ovat myös vakaita, tietoturvallisia ja niihin löytyy runsaasti ohjeita sekä Internetistä että kirjoista.

4.2.1 Microsoft Visual Studio 2010

Microsoft Visual Studio 2010 on Microsoftin tuottama sovelluksien kehittämisympäristö. Työkalu on julkaistu tuotenimensä mukaisesti vuonna 2010. Visual Studiossa tuetut pääohjelmointikielet ovat Visual Basic, Visual C#, Visual C++, Visual F# ja JavaScript. Näistä kielistä kaikilla muilla paitsi JavaScriptillä on mahdollista käyttää .NET-ohjelmistokomponenttikirjastoa. Työkalulla on helppoa tehdä myös graafisia käyttöliittymiä sovelluksiin. Tietyissä sovelluksissa on mahdollista käyttää yhtä aikaa myös muita kieliä, joita ovat esimerkiksi HTML, XHTML ja CSS. Työkalulla voi kehittää työpöytä-, WWW-, palvelu- ja mobiilisovelluksia. [19,20]

Työpöytäsovellukset ovat sovelluksia, jotka ladataan Internetistä tai asennetaan paikallisesti esimerkiksi USB-muistitikulta tai CD-levyltä. Sovellukset luokitellaan usein seuraaviin ohjelmaluokkiin: tekstinkäsittelyohjelmat, henkilökohtaiset talousohjelmat, WWW-selaimet, pelit, mediatoistimet ja pienoisojgelmat. Työpöytä-

sovelluksien suhteellinen määrä on laskenut, sillä Internet-yhteyksien yleistyessä esimerkiksi WWW-ohjelmien suosio on kasvanut. [21]

WWW-sovellus on WWW-selaimen kautta käytettävä sovellus. Sovellus pyörii yleensä erillisellä palvelimella, josta asiakassovellus hakee tietoja. WWW-sovelluksien käyttämisen helppous on niiden suosion salaisuus, sillä WWW-sovellukset eivät tarvitse erillistä asennettavaa ohjelmaa, vaan ainoastaan WWW-selaimen, joka on nykyään vakiovaruste melkein kaikissa päätelaitteissa. [18]

Palvelusovellukset ovat pitkäaikaiseen käyttöön tarkoitettuja sovelluksia. Ne käynnistyvät tietokoneen kanssa samaan aikaan, mikä tarkoittaa sitä, että ne ovat käytössä ennen kuin käyttäjä kirjautuu tietokoneeseen. Niillä ei ole näkyvää käyttöliittymää vaan sovellukset pyörivät taustalla taustasovelluksina. Taustasovellukset eivät myöskään häiritse käyttäjää mitenkään, koska ne eivät anna ilmoituksia vaan niiden toiminnasta saa tietoa vain esimerkiksi viemällä virheilmoitukset lokitiedostoon. Palvelusovelluksia käytetään useimmiten palvelinkoneilla, jotka ovat päällä käytännössä koko ajan ja joissa ei välttämättä ole käyttäjää kirjautuneena sisään. Palvelusovelluksien kehittäminen on hieman hitaampaa kuin normaalien sovelluksien kehittäminen, sillä niitä ei pysty ajamaan suoraan kehitystyökalulla vaan ne täytyy asentaa joka kerta erikseen ja sitten vasta voidaan testata niiden toimintaa. Asennetulla palvelusovelluksella on kolme eri tilaa: käynnissä, pysäytetty ja lopetettu. [22]

Mobiilisovellukset ovat pienille ja langattomille laitteille tarkoitettuja sovelluksia. Kohdelaitteita ovat esimerkiksi älypuhelimet. Sovelluksia voidaan käyttää vaihtoehtona nettisivuille, jotka eivät ole responsiivisia eli laitteen näytölle skaalautuvia. Niihin voidaan laittaa vain sellaista sisältöä, jota mobiilikäytössä tarvitaan. Mobiilisovellukset luokitellaan usein kolmeen luokkaan. Nämä luokat ovat natiivi mobiilisovellus, HTML5 WWW-sovellus ja hybridi mobiilisovellus. Natiivi mobiilisovellus on kallis ja hankala tapa kehittää mobiilisovelluksia isoille käyttäjäkunnille, sillä sovellus täytyy kehittää jokaiselle laitealustalle erikseen. Natiivi mobiilisovellus pystyy käyttämään laitealustan kaikkia toimintoja ja rajapintoja, mikä parantaa sen suorituskykyä. HTML5-WWW-sovellus on puolestaan laitealustas-

ta riippumaton sovellus, joka soveltuu erityisesti mobiiliratkaisuihin. HTML5 - WWW-sovellus toimii WWW-selaimen kautta, mikä mahdollistaa sovelluksen helpon päivittämisen. Hybridi mobiilisovellus on natiivin mobiilisovelluksen ja HTML5-WWW-sovelluksen yhdistelmä, joka käyttää molempien menetelmien parhaita puolia. Hybridi mobiilisovelluksen käyttäminen on nopeaa, päivityskustannukset ovat pienet ja sovellus pystyy hyödyntämään laitteen toimintoja ja rajapintoja. [23]

4.2.2 Microsoft SQL Server 2008

Microsoft SQL Server on Microsoftin tuottama tietokantapalvelinjärjestelmä. Työkalu on julkaistu tuotenimensä mukaisesti vuonna 2008. Sen tarkoituksena on toimia tietokantapalvelimenä muille sovelluksille. Tietokantapalvelimelta tietoa hakevat sovellukset toimivat asiakassovelluksina, jotka voivat sijaita samassa laitteessa tai verkkoyhteyden päässä. [24]

Tietokantapalvelinjärjestelmään voidaan luoda ja hallita tietokantoja kahdella eri tavalla. Ensimmäisessä tavassa käytetään ulkopuolista asiakasohjelmaa. Toisessa tavassa käytetään Microsoftin omaa hallintasovellusta, jonka nimi on Microsoft SQL Server Management Studio. [25]

Käytettäessä ulkopuolista asiakasohjelmaa, joudutaan konfiguroimaan enemmän asioita, jotta tietokantapalvelinta pystytään käyttämään. Tärkein konfigurointi liittyy tietokantayhteyden luomiseen tietokantapalvelimen ja asiakasohjelman välille. Luodessa tietokantayhteyden verkkoyhteyden yli, joudutaan luomaan tietokantapalvelimelle käyttäjätunnus ja salasana, mitkä syötetään asiakasohjelmaan. [26]

Microsoft SQL Server Management Studio on Microsoftin kehittämä hallintajärjestelmä, jonka avulla voidaan helposti hallita, konfiguroida ja ylläpitää SQL Serverillä pyörivää tietokantaa. Sovellukset ovat täysin yhteensopivia toistensa kanssa. SQL Server Management Studio sisältää myös helppokäyttöisiä graafisia työkaluja, jotka mahdollistavat hallintajärjestelmän sujuvamman käytön eritasoisilta kehittäjiltä ja ylläpitäjiltä. [25]

4.3 Kehitysmenetelmät

Käytettyihin kehitysmenetelmiin kuului .NET-ohjelmistokomponenttikirjasto ja ASP.NET-WWW-ohjelmistokehys. .NET-ohjelmistokomponenttikirjasto on katavuuden, helppouden ja monien ohjelmointikielien tuen vuoksi sopiva kehitysmenetelmä moniin eri ohjelmistoprojekteihin. ASP.NET-WWW-ohjelmistokehys kuuluu .NET-ohjelmistokehykseen, joten sen kautta voidaan käyttää .NET-ohjelmistokomponenttikirjaston tarjoamia etuja. WWW-sovelluksen kehitys tällä ohjelmistokehyksellä on myös helposti opittava kehitysmenetelmä.

4.3.1 .NET-ohjelmistokomponenttikirjasto

.NET-ohjelmistokomponenttikirjasto on Microsoftin tuottama ohjelmistokehitykseen tarkoitettu komponenttikirjasto. Sen pääasiallinen käyttäminen tapahtuu Microsoft Visual Studio –ohjelmistokehitysympäristössä. Komponenttikirjasto tukee esimerkiksi Visual Basic, Visual C#, Visual F# ja Visual C++ –ohjelmointikieliä. Sen käyttäminen edellyttää Microsoftin NT –käyttöjärjestelmää, joita ovat muun muassa Windows Vista ja Windows 7. [27]

.NET-ohjelmistokomponenttikirjaston käyttäminen perustuu luokkakirjastojen ja ajoympäristön yhteistoimintaan. Käyttäjän kirjoittaman koodin määrä vähenee, sillä yhdistelmä suorittaa paljon valmiiksi luotuja koodisarjoja. Tämä auttaa ohjelmoijaa keskittymään enemmän toiminnallisuuden ohjelmointiin. Ohjelmointi tapahtuu käytännössä hakemalla komponenttikirjastosta jokin osatoiminnallisuus vain parilla komennolla ja muokkaamalla koodiin omat parametrit. Valmiiksi luoduissa koodisarjoissa on myös huomioitu muistinhallinta, joka täytyisi muuten itse käsitellä ohjelmointikoodissa. Muistinhallintaa voidaan kuitenkin joutua käsittelemään monimutkaisissa ohjelmointiprojekteissa. [27]

Ajoympäristö kääntää ohjelmointikielestä riippumattoman esikäännetyn ohjelmointikoodin binäärikoodiksi eli natiivikoodiksi, jota käyttöjärjestelmä ymmärtää. Ajoympäristön käyttäminen vähentää yhteensopivuusongelmia eri suoritintyyppien välillä. Ajoympäristön käyttäminen sovelluksen ajossa on hitaampaa kuin

valmiiksi käännetyin binäärikoodin ajaminen, sillä ajonaikainen kääntäminen tuo yhden välivaiheen lisää sovelluksen suorittamiseen. Tämä tarkoittaa myös sitä, että esikäännettyä ohjelmointikoodia sisältävän ohjelman lähdekoodia on erittäin vaikeaa salakirjoittaa. [28]

.NET-ohjelmistokomponenttikirjasto tukee seuraavia sovelluksia: konsolisovellukset, graafiset sovellukset, WWW-sovellukset, palvelusovellukset ja mobiilisovellukset. Komponenttikirjastosta löytyviä ohjelmistokomponentteja ovat esimerkiksi käyttöliittymä-, tietokanta-, salakirjoitus-, verkko-, WWW- ja algoritmi-komponentit. Komponenttikirjaston käyttäminen on hyödyllistä aloitteleville ohjelmoijille, sillä sitä käyttämällä pystytään helposti tekemään toimivia sovelluksia pienemmällä määrällä ohjelmointikoodia. [27,28]

4.3.2 ASP.NET-WWW-ohjelmistokehys

ASP.NET-WWW-ohjelmistokehys on Microsoftin tuottama dynaamiseen WWW-ohjelmointiin tarkoitettu ohjelmistokehys. Ohjelmistokehys kuuluu .NET-ohjelmistokehykseen, joten ohjelmistokehystä käyttäessä pystyy käyttämään hyväkseen .NET-ohjelmistokomponenttikirjaston luokkia. Ohjelmistokehyksen avulla ohjelmoija pystyy luomaan WWW-sivuja, WWW-sovelluksia ja WWW-palveluja. ASP.NET-WWW-ohjelmistokehys pohjautuu Microsoftin ASP-ohjelmointimenetelmään. Ohjelmistokehys käyttää .NET-ajoympäristöä, joka mahdollistaa ohjelmointikoodin kirjoittamisen tuetuilla .NET-ohjelmointikielillä, joita ovat esimerkiksi Visual C#, Visual Basic, JScript .NET ja Visual J#. [29]

ASP.NET-WWW-ohjelmistokehyksellä ohjelmoidut WWW-sivut, WWW-sovellukset ja WWW-palvelut toimivat WWW-palvelimella, josta niitä pystyy hakemaan oletusarvoisesti millä tahansa WWW-selaimella tai päätelaitteella. Ohjelmistokehyksellä on myös mahdollista kohdentaa ohjelmistokehitys ainoastaan tietyille WWW-selaimelle, jolloin ohjelmistokehityksessä voi hyödyntää valitun WWW-selaimen etuja. Ohjelmistokehys tukee monia mobiilikäyttöön tarkoitettuja ohjelmistokomponentteja. Ohjelmistokehitys on vahvasti olio-ohjelmointimenetelmään keskittynyttä tässä ohjelmistokehyksessä. [29]

ASP.NET-WWW-ohjelmistokehys on jaettu kolmeen aliohjelmistokehykseen: ASP.NET Web Forms, ASP.NET MVC ja ASP.NET Web Pages. Jokaisella aliohjelmistokehyksellä on omat vahvuutensa ja käyttötarkoituksensa. Ohjelmistokehysten valinta perustuu WWW-ohjelmointi kokemukseen, ohjelmistokehysten tuntemukseen ja käyttötarkoitukseen sopivuuteen. On myös mahdollista, että samassa WWW-ohjelmistossa on käytetty monia eri ohjelmistokehyksiä, mikä tarkoittaa sekamuoto-ratkaisua. Sekamuoto-ratkaisussa käytetään hyväksi eri ohjelmistokehysten parhaita puolia. [30]

ASP.NET Web Forms –WWW-ohjelmistokehys sopii ohjelmistokehittäjille, joilla on ohjelmointikokemusta esimerkiksi WinForms tai WPF/XAML/Silverlight –ohjelmointimenetelmistä. Ohjelmistokehys sopii nopeaan WWW-ohjelmistokehitykseen, sillä se tukee vedä ja pudota –ohjelmistokehitysmallia. Siirtyminen WWW-ohjelmistokehitykseen ASP.NET Web Forms –WWW-ohjelmistokehysten kautta on helppoa ohjelmistokehittäjille, joilla on kokemusta nopean ohjelmistokehityksen ohjelmointikielistä, sillä tämän WWW-ohjelmistokehysten käyttäminen ei vaadi asiantuntemusta HTML tai JavaScript –kielistä. Tällä WWW-ohjelmistokehyksellä WWW-ohjelmistokehitys on nopeampaa ja vähemmän ohjelmointia vaativaa kuin esimerkiksi käytettäessä ASP.NET MVC –WWW-ohjelmistokehystä. ASP.NET Web Forms –WWW-ohjelmistokehysten nopealla ohjelmistokehitysmahdollisuudella on myös haittapuolensa, sillä HTML-kielen ja HTTP-protokollan käyttäytymistä WWW-ohjelmistossa ei pystytä tarkkaan määrittelemään. [30]

ASP.NET MVC –WWW-ohjelmistokehys sopii ohjelmoijille, jotka haluavat erottaa toiminnallisuuskoodin ja käyttöliittymäkoodin toisistaan. Ohjelmistokehysten nimessä esiintyvä MVC on lyhenne sanoista model, views ja controllers. Lyhenne tarkoittaa sitä, että WWW-ohjelmistokehitys on jaettu lyhenteessä esiintyviin osiin eli malleihin, näkymiin ja ohjaimiin. Ohjelmistokehityksen jakaminen osiin helpottaa laajojen ja monimutkaisten ohjelmistojen kehitystä. Osiin jakamisen ansiosta myös tiimityöskentely helpottuu, sillä osa sovelluskehitystiimistä voi kehittää toiminnallisuuskoodia ja osa voi työskennellä käyttöliittymäkoodin parissa. ASP.NET MVC –WWW-ohjelmistokehysten etuja on esimerkiksi se, että

HTML-kielen ja HTTP-protokollan käyttäytymisen WWW-ohjelmistossa pystyy määrittelemään tarkasti. [30]

ASP.NET Web Pages –WWW-ohjelmistokehyksessä luodaan ensin WWW-sivuja HTML-koodilla ja sen jälkeen lisätään palvelinpohjaista koodia. Ohjelmistokehystä on helppo käyttää, jos osaa HTML-kieltä vaikka ei muuten ole vahvaa ohjelmointikokemusta. PHP-WWW-ohjelmistokehyksen tuntemuksesta on myös hyötyä siirryttäessä käyttämään ASP.NET Web Pages –WWW-ohjelmistokehystä. Ohjelmistokehys sopii nopeaan WWW-ohjelmistokehitykseen, sillä se sisältää paljon valmiita ohjelmistokomponentteja. [30]

4.4 Kielet

ASP.NET-WWW-ohjelmistokehyksen valinnan avulla ohjelmistoprojektissa pystyi käyttämään viittä eri kieltä. Nämä kielet olivat Visual C#, XHTML, CSS, JavaScript ja SQL. Ohjelmistokehityksen osa-alueiden jakaminen eri kielille auttoi pilkkomaan suunnitteluakin pienempiin osiin. Jokaisella kielellä on oma vahvuutensa, jota pystyi hyödyntämään sopivissa paikoissa. SQL-hakukielellä keskityttiin luonnollisesti hakemaan tietoja tietokannasta. Visual C# ja JavaScript – kielillä toteutettiin WWW-sovelluksen toiminnallisuudet. CSS-kielellä toteutettiin tyyliohjeisiin liittyvät seikat, jolloin saman tyylin tiedot haettiin aina erillisestä tyyliohjetiedostosta. XHTML-merkintäkielellä toteutettiin WWW-sovelluksen graafinen puoli, johon kuului esimerkiksi kaavioiden piirtäminen tietokannasta haettujen tietojen perusteella.

4.4.1 Visual C#

Visual C# on Microsoftin tuottama ohjelmointikieli. Kieli suunniteltiin .NET-ohjelmistokomponenttikirjastoa varten. Kieli suunniteltiin myös käytettäväksi monissa ohjelmointiympäristöissä, tukemaan olio-ohjelmointia ja olemaan helpposti kansainvälistettävä. Kieli on yhdistelmä C, Java ja Visual C++ -

ohjelmointikieliä. Visual C# noudattaa C:n syntaksia. Kielen helppokäyttöisyys perustuu Javaan. Kielessä on käytössä Visual C++:n monipuolisia ja tehokkaita ohjelmointitekniikoita, jotka on tehty yksinkertaisemmiksi käyttää. Kielen helppokäyttöisyyden ja nopean opittavuuden vuoksi se sopii mainiosti nopeaan ohjelmistokehitykseen. [31]

Visual C#:ssa on painotettu muuttujille annettavien muuttujatyypin määrittämisestä. Erilaisia muuttujatyyppejä ovat esimerkiksi merkkijono, kokonaisluku, desimaaliluku ja totuusarvo. Helppokäyttöisyyttä edistää muuttujien alustamisen automaattinen tarkistus ja muuttujien automaattinen kuolettaminen muistinkäytön optimoimiseksi. Kielellä ei kuitenkaan pysty tekemään yhtä optimoituja sovelluksia kuin esimerkiksi C:llä tai Visual C++:lla. [31]

Visual C#:lla ohjelmoidun sovelluksen lähdekooditiedostot on nimetty yhteensopivasti muiden Microsoftin kehittämien ohjelmointikielten mukaisesti. Visual C#:n lähdekooditiedostoissa on ".cs"-tiedostopääte, kun taas esimerkiksi C:ssä on ".c"-tiedostopääte ja Visual C++:ssa on ".cpp"-tiedostopääte. Visual C#:lla, C:lla ja Visual C++:lla käännettyjen ajettavien tiedostojen tiedostopäätteet ovat samanlaiset, joista tunnetuimmat ovat .exe ja .dll. Microsoftin Visual C#, C ja Visual C++ -ohjelmointikielillä on paljon yhteensopivia menetelmiä, sillä esimerkiksi Visual C#:lla on mahdollista käyttää Visual C++:lla tehtyä jaettua ohjelmointikielikirjastoa. Visual C# -ohjelmointikieleen on helppo siirtyä jos tuntee C, Java tai Visual C++ -ohjelmointikieliä. Siirtymisen tekee helpoksi se, että ohjelmointikoodin kirjoittamisessa on yhteneväisyyksiä, joita ovat esimerkiksi puolipisteet, aaltosulut, muuttujien arvojen määritykset, muuttujien arvojen vertailut ja taulukoiden käyttäminen. [31,32]

Visual C#:lla ohjelmoitu ajettava tiedosto sisältää esikäännettyä ohjelmointikoodia. Tiedoston ajaminen vaatii ajoympäristön, jonka avulla esikäännetty ohjelmointikoodi käännetään binäärikoodiksi, jota käyttöjärjestelmä ymmärtää ja osaa suorittaa halutut toiminnot. Erillisen ajoympäristön tarvitseminen parantaa yhteensopivuutta mutta pidentää ajettavan tiedoston ajoaikaa. Visual C#:lla ohjelmoidun ajettavan tiedoston lähdekoodin pystyy helposti kääntämään suoraan

takaisin ohjelmointikoodiksi. Ajettavan tiedoston lähdekoodin helppo saatavuus vaikuttaa ohjelmointikielen valintaan etenkin yrityskäytössä. [31]

4.4.2 XHTML

XHTML on avoin WWW-sivujen merkintäkieli, joka on kehitetty tukemaan XML-kieltä. Kieli muistuttaa hyvin läheisesti HTML-merkintäkieltä. XHTML- ja HTML-merkintäkielet eroavat toisistaan kielessä käytettävän muotoilun eroilla. XHTML-merkintäkielessä on tiukemmat muotosäännöt kuin HTML-merkintäkielessä. XHTML-merkintäkielen ensimmäinen versio julkaistiin nimellä XHTML 1.0 ja tällä hetkellä uusin versio on nimeltään XHTML5. [33]

XHTML-merkintäkielellä luodaan käytännössä WWW-sivuja yksinkertaisimmillaan tekstinkäsittelyohjelman avulla. Merkintäkielellä merkitään, mikä osa tekstistä on esimerkiksi ylätunnistetta, alatunnistetta, päättekstiä, taulukoita, listoja, tai kuvia. Tekstin osat rakentuvat tunnisteista, tunnisteiden ominaisuuksista ja tunnisteiden sisällöistä. Tekstin osat voivat olla joko sisäkkäin tai peräkkäin. [33]

Vaikka XHTML- ja HTML-merkintäkielet ovat hyvin samanlaisia, niillä on myös eroja, jotka määrittelevät, kumpaa merkintäkieltä kannattaa käyttää. XHTML-merkintäkielen tärkeimmät edut ovat seuraavat: tunnisteiden laajennettavuus erikseen jokaiselle eri WWW-sivulle, XML-kielen tuki, tehtyjen dokumenttien helpompi hallittavuus, helpompi opittavuus, helpompi opetettavuus ja pidempi yhteensopivuus uudempiin versioihin. HTML-merkintäkielen etu on suurempi suosio verrattuna XHTML-merkintäkieleen. [33]

Siirtyminen HTML-merkintäkielen käyttämisestä XHTML-merkintäkielen käyttämiseen on helppoa, koska ne ovat samankaltaisia. XHTML-merkintäkielen tiukemmista muotosäännöistä johtuen, XHTML-merkintäkieltä on helpompi muuttaa HTML-merkintäkielen kanssa yhteensopivaksi kuin toisinpäin. Muutettaessa HTML-dokumenttia XHTML-dokumentiksi, täytyy tehdä seuraavat muutokset tai lisäykset: alkuun pitää määritellä dokumenttityyppi, tunnisteet täytyy kirjoittaa pienillä kirjaimilla kuten myös niiden ominaisuudetkin, tunnisteiden sulkeminen tapahtuu sääntöjen mukaisella tavalla, jokaisen ominaisuuden arvo täytyy olla

lainausmerkkien sisällä, ominaisuuksien lyhentäminen ei ole sallittua, identiteetti-ominaisuudet korvaavat nimi-ominaisuudet, kieli-ominaisuutta ei käytetä sekä jokaisen tunnistekokonaisuuden alku- ja lopputunnisteet täytyy olla sisäkkäin toista vastaavalla paikalla. [33]

4.4.3 CSS

CSS on tyyliohjekieli, jolle luodaan tyyliohjetiedosto. Tyyliohjetiedostosta voidaan hakea tyyliohjeita moneen eri paikkaan. Ajatuksena on erottaa sisältö ja sen esittäminen toisistaan. Käytettäessä CSS:ää esimerkiksi WWW-sivujen kehityksessä, jokaiselle eri sivulle ei tarvitse määritellä käytettäviä tyyliä vaan yhtenevät tyylit voi hakea suoraan erillisestä tyyliohjetiedostosta. CSS:n käytämisestä on tullutkin suosittua etenkin WWW-kehittäjien keskuudessa, johtuen myös siitä, että melkein kaikki WWW-selaimet tukevat CSS:n käyttöä nykyään. CSS:n ensimmäinen versio julkaistiin nimellä CSS1 ja tällä hetkellä uusin versio on nimeltään CSS3. [34]

CSS-tyyliohjekielen runko koostuu kolmesta osasta. Ensimmäinen osa on valittava kohde, jolla määritellään mihin muuttujaan tyylimääritys vaikuttaa. Tämän jälkeen tulee aaltosulkeiden sisään kaksi muuta osaa, jotka erotellaan puolipisteellä. Näistä aaltosulkeiden sisällä olevista osista ensimmäisellä määritellään ominaisuus ja toisella osalla annetaan ominaisuudelle haluttu arvo. [34]

CSS-tyyliohjekielellä voidaan vaikuttaa esimerkiksi asioiden sommitteluun, väriin, fontteihin, korostuksiin, korkeuksiin ja leveyksiin. Näiden lisäksi CSS:llä pystyy helpottamaan esimerkiksi responsiivisten WWW-sivujen toteutusta, sillä CSS:llä voidaan määritellä miltä WWW-sivu näyttää milläkin laitteella tai näytönkokoalla. Tyyliohjeisiin voidaan myös määritellä kohteen tulostusasetukset. Tyyliohjetiedostoa käyttämällä vähennetään todennäköisyyttä, että jollakin kohteella on tahtomattaan kaksinkertaiset tyyliohjeet eri paikoissa. [34]

4.4.4 JavaScript

JavaScript on WWW-ohjelmistokehityksessä käytettävä ohjelmointikieli, jonka on luonut Netscape Communications Corporation. Kielen vastuullinen kehittäjä on tällä hetkellä ECMA-järjestö. Kielen tarkoituksena on usein toimia asiakaspuolen toiminnallisuuksien toteuttajana. Ideana on tuottaa WWW-sivuilla vierailvalle käyttäjälle dynaamista sisältöä. Tämä tarkoittaa sitä, ettei WWW-sivua tarvitse ladata palvelimelta käyttäjälle uudestaan jos halutaan muokata käyttäjän näkymää esimerkiksi hyödyntämällä käytettävän selaimen ominaisuuksia, käyttäjän suorittaessa ennalta määritelty toiminto tai muutettaessa WWW-selaimen ikkunan kokoa. JavaScriptiä käytetään usein WWW-selaimissa ajettavien toiminnallisuuksien toteutukseen. Asiakaspuolella pyörivä koodi luonnollisesti vähentää palvelimien kuormitusta, sillä esimerkiksi lomakkeisiin täytettyjen tietojen validointi voidaan suorittaa jo ennen kuin palvelin saa tiedot ja näin pystytään antamaan mahdollista palautetta käyttäjälle vaikka palvelimella olisikin ruuhkaa. JavaScriptillä voidaan myös luoda käyttäjän tietokoneelle evästeitä, joilla voidaan muokata käyttäjän seuraavaa WWW-selauskokemusta entistä persoonallisemmaksi. [35]

JavaScript-ohjelmointikielen nimi on ollut alun perin LiveScript. Nimen vaihtaminen on saanut monissa ihmisissä aikaan mielikuvan, että JavaScript ja suosittu ohjelmointikieli Java olisivat hyvin lähellä toisiaan asiassa kuin asiassa. Kielet kuitenkin eroavat paljon toisistaan, vaikka niiden käyttämissä ohjelmistokomponenttikirjastoissa onkin joitakin samanlaisuuksia ja molempien kielien muoto säännöt pohjautuvat Microsoftin kehittämään C++-ohjelmointikieleen. JavaScriptin kasvavasta suosiosta johtuen Microsoft tuotti lähelle menevän ohjelmointikielen ja antoi sille nimeksi JScript. Tämä nimi on saanut aikaan sekaannuksia, muttei ole rikkonut JavaScriptin tavaramerkkioikeuksia, jotka omistaa Oracle Corporation. [35,36]

JavaScriptin käyttäminen tarkoittaa yksinkertaisimmillaan sitä, että koko kieltä ei tarvitse välttämättä opetella ollenkaan, vaan JavaScript-palikoita voidaan kopioida suoraan esimerkiksi omille WWW-sivuille, jolloin täytyy ainoastaan tietää

mihin kopioitu koodi liitetään. JavaScript-koodia voidaan kehittää joko erilliseen tiedostoon tai WWW-sivuja kehitettäessä suoraan HTML-koodin sekaan skriptitunnisteilla varustettuna. Kehitettäessä omia JavaScript-tiedostoja, työkaluksi käy esimerkiksi normaali tekstinkäsittelyohjelma, jolla koodi tallennetaan ".js"-tiedostopäätteellä. [35]

JavaScriptiä käytetään useimmiten asiakaspuolella mutta siitä on tehty myös palvelinpuolella toimivia ratkaisuja, joita on esimerkiksi "Node.js". Palvelinpuolen ratkaisuja on kehitetty asiakaspuolen rinnalle, koska normaalisti JavaScript-koodia ajetaan käyttäjän selaimessa, mikä tarkoittaa sitä, että selaimessa täytyy olla käytössä JavaScript-tuki. JavaScriptin suosiosta johtuu, että sitä tukevat suosituimmat WWW-selaimet ja näissä selaimissa on JavaScript-tuki päällä automaattisesti. JavaScript-koodit eivät välttämättä kuitenkaan toimi käyttäjällä jos esimerkiksi JavaScript on otettu pois käytöstä turvallisuussyistä tai päätelaitteen WWW-selain ei tue JavaScriptiä. Tämän takia WWW-sivut olisi hyvä toteuttaa mahdollisimman hyvän saavutettavuuden takia niin, että pakolliset toiminnot toimivat selaimesta riippumatta ja ylimääräiset toiminnot käyttävät asiakaspuolella esimerkiksi JavaScriptiä. Tällainen WWW-kehitys sopii tärkeille WWW-palveluille, joita ovat esimerkiksi terveyspalvelut. Asiakaspuoleen keskitetty JavaScript tarvitsee useimmiten lisäksi muutakin koodia, sillä esimerkiksi tietokantoja ei ole mahdollista käyttää JavaScriptin avulla. [35]

4.4.5 SQL

SQL on relaatiotietokantojen hallintaan tarkoitettu kyselykieli. SQL tulee sanoista Structured Query Language. Kielen on luonut alun perin IBM. Kielen avulla pystytään hakemaan, muuttamaan, poistamaan ja lisäämään tietoja, joita voivat olla koko tietokanta, taulu, näkymä tai tietoalkio. SQL-kyselykieli kuuluu ANSI- ja ISO-standardeihin. SQL-kieltä tukevat melkein kaikki relaatiotietokantojen hallintaohjelmistot, joita ovat esimerkiksi Microsoft SQL Server, MySQL, Oracle, IBM DB2 ja PostgreSQL. [16,37]

SQL-kyselykieli jaotellaan usein kolmeen kieliluokkaan käyttötarkoituksen mukaan. Luokat ovat tietojen määrittelykieli, tietojen käsittelykieli ja tietojen hallintakieli. Tietojen määrittelykielessä luodaan, muokataan ja poistetaan määrittelykohteita, joita ovat esimerkiksi tietokannan taulut. Määrittelykielen pääkäskyt ovat CREATE, ALTER, TRUNCATE ja DROP. Seuraavassa kieliluokassa, tietojen käsittelykielessä, käsitellään tietokannan taulujen tietueita. Tämän kieliluokan pääkäskyt ovat SELECT, INSERT, UPDATE ja DELETE. Tietojen hallintakielessä määritellään tietokannan eri osien hallintaoikeuksia eri käyttäjille. Hallintakielen pääkäskyt ovat GRANT ja REVOKE. SQL-kyselykielen laajuudesta huolimatta, mainittujen kolmen kieliluokan pääkäskyjen avulla selviää useimmista relaatiotietokantojen hallintaan liittyvistä tehtävistä. [37]

SQL-kyselykielen muotosäännöt koostuvat kyselylausekkeista. Yksittäinen kyselylauseke pitää sisällään lauseita, jotka alkavat englanninkielisellä sanalla. Englanninkielisen sanan jälkeen tulee muuttuja, jolle määritellään yhtäsuuruusmerkin jälkeen tarkasteltava arvo. Yksinkertaisimmillaan kyselylauseke voi koostua vain yhdestä lauseesta mutta monimutkaisimmillaan kyselylauseke pitää sisällään myös monia alikyselylausekkeitä, jolloin lauseiden määrä kasvaa moninkertaiseksi. [37]

4.5 Toteutus

Opinnäytetyön toimeksiantajalla oli suunnitteilla ihmisten kuntoutusta parantava projekti. Projektin tarkoituksena oli tuottaa järjestelmä, jossa kuntoutettava tekee kuntoutusliikkeitä ja liikkeet tallentuvat sähköiseen tietokantaan. Tämän jälkeen tietokannasta oli tarkoitus hakea haluttuja tietoja, jotka kuntouttaja ja kuntoutettava näkisivät WWW-selaimen kautta omista näkymistään. Opinnäytetyö rajattiin koskemaan prototyyppiä, johon kuului tietokannan ja WWW-sovelluksen kehittäminen. Prototyypistä tehtiin aluksi vaatimusmäärittely, jossa määriteltiin esimerkiksi jokaisesta kuntoutettavasta ihmisestä tallennettavia perustietoja ja kuntoutukseen liittyviä tietoja. Tietokantaa hyödyntävän WWW-sovelluksen kehittäminen piti sisällään kaksi teknistä osaa. Ensimmäinen tekni-

nen osa koostui tietokannan suunnittelusta ja toteutuksesta. Toisessa osassa suunniteltiin WWW-sovellus ja yhdistettiin siihen aiemmin luotu tietokanta.

4.5.1 Tietokannan luominen

Tietokannan hahmottelu alkoi jo opinnäytetyön aloituspalaverissa. Palaverissa päätettiin, mikä on opinnäytetyön tarkoitus ja mitä siltä odotetaan. Tietokannan suunnittelu alkoi konkreettisesti tämän jälkeen kynällä, paperilla ja tietokoneella. Ensimmäiseksi suunniteltiin tietokannan tietoalkioita ja sitä, mitä tietoja tarvittiin ja mitä tietotyyppejä tiedot ovat. Tietokannan tietoalkioista oli jo selvillä suurin osa. Ne olivat suoraan pääteltävissä tehdystä vaatimusmäärittelystä mutta asiasta päätettiin lisäksi konsultoida fysioterapeuttia, jolla on pitkä kokemus terveydenhoitopuolelta. Tietokannan tietoalkioiden suunnittelusta päästiin suoraan siihen, miten tiedot liittyvät toisiinsa, minkä tietoalkioiden välillä vallitsee suora tietokantarelaatio ja millaisia tauluja tietokantaan on tarkoitus luoda. Näiden esitietojen toimeksiantajalla hyväksyttämisen jälkeen suunnitteleminen kohdistui tietokannan hallintaohjelmiston valintaan. Valinta osui loppujen lopuksi Microsoftin SQL Serveriin. Valintaan vaikutti suurimmaksi osaksi se, että WWW-ohjelmisto oli suunniteltu toteutettavaksi Microsoftin Visual Studiolla, jonka kanssa SQL Server toimii saumattomasti yhteen. Tietokannan luonti toteutettiin Visual Studiosta löytyvällä graafisella työkalulla (Kuva 2.). Tietokantaan luotiin aiemmin suunnitellut taulut ja jokaiseen tauluun kuuluvien solujen tietotyypit. Tietokannan toimivuuden parantamiseksi luotujen taulujen välille luotiin relaatiot, joista pystytään yhdistämään helposti toisiinsa kuuluvat tiedot (Kuva 3.). Näiden työvaiheiden jälkeen tietokantaan liittyvät asiat oli saatu päätökseen.

Add Connection

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:
Microsoft SQL Server (SqlClient) Change...

Server name:
▼ Refresh

Log on to the server

☒ Use Windows Authentication

☐ Use SQL Server Authentication

User name:

Password:

☐ Save my password

Connect to a database

☒ Select or enter a database name:
▼

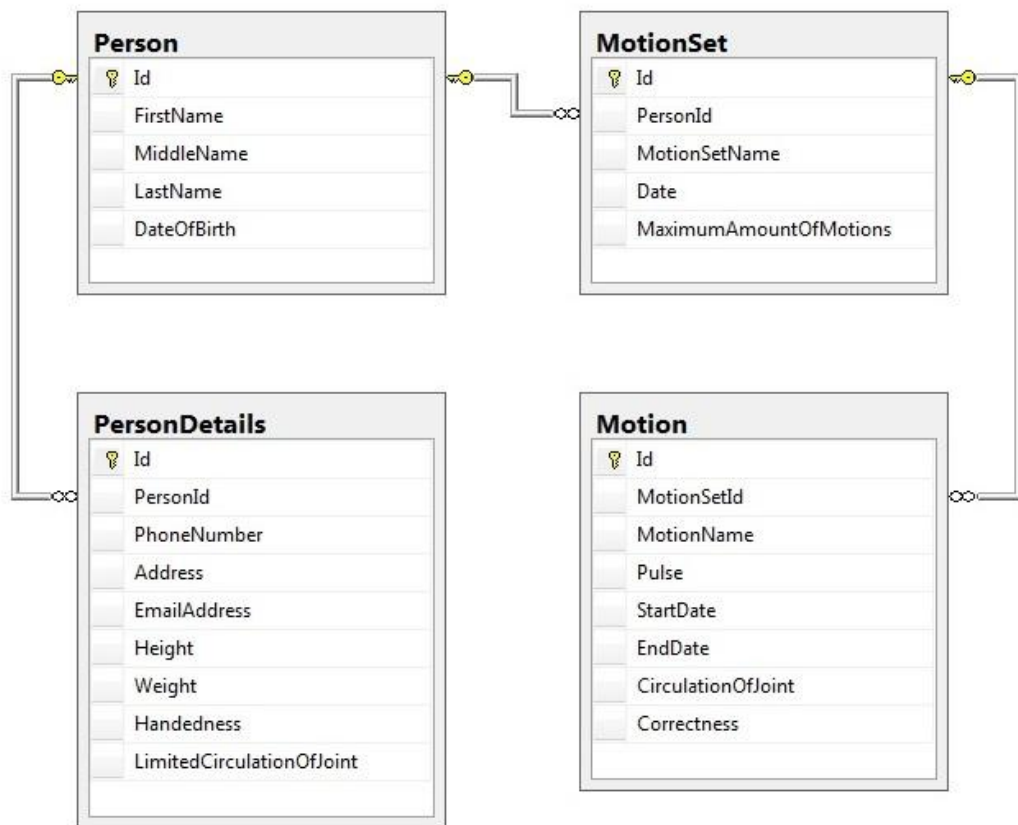
☐ Attach a database file:
 Browse...

Logical name:

Advanced...

Test Connection OK Cancel

Kuva 2. Tietokannan luonti Microsoft Visual Studio graafisella työkalulla.



Kuva 3. Tietokannan taulujen relaatiot.

4.5.2 WWW-sovelluksen kehittäminen

WWW-sovelluksen suunnittelu alkoi samassa vaiheessa, kun tietokantasuunnittelussa pohdittiin tarvittavia tietoalkioita ja niiden esitystapaa. Samalla mietittiin käyttötarkoitukseen sopivaa kehitystyökalua. WWW-sovelluksen kehittämisestä ei ollut aiempaa kokemusta, joten tutun kehitystyökalun valinnalla WWW-sovelluksen kehittäminen oli helpompi aloittaa. Kehitystyökaluksi valittiin tällä perusteella Microsoftin Visual Studio. Tämän jälkeen oli vuorossa WWW-ohjelmistokehyksen valinta, joka kohdistui ASP.NET MVC –WWW-ohjelmistokehykseen, sillä sen ominaisuudet vastasivat parhaiten käyttötarkoitusta. WWW-ohjelmistokehyksen valinnan takia käytettäviksi kieliksi tulivat Visual C#, XHTML, CSS, JavaScript ja SQL. Käyttämällä monia eri kieliä pystyttiin jakamaan kullekin kielelle oma osa-alueensa, ja näin sovelluksen suunnittelua kin pystyttiin pilkkomaan pienempiin osiin. WWW-sovelluksen hahmotelma hy-

väksytettiin toimeksiantajalla yhtä aikaa kuin tietokantahahmotelma. Tämän jälkeen oli järkevintä tehdä tietokanta loppuun ja keskittyä sen jälkeen muihin WWW-sovelluksen osa-alueisiin. Nämä osa-alueet olivat toiminnallisuuden ja graafisen käyttöliittymän kehittäminen. Toiminnallisuuden kehittäminen alkoi tietokantahakujen ja eri näkymien välillä siirtymisen suunnittelulla. Kokonaiskuvan hahmotteluun käytettiin tarpeeksi aikaa, jotta aikaa vieviä virheitä tulisi mahdollisimman vähän jälkimmäisissä kehitysvaiheissa. Kokonaiskuvan hahmotteluun kuului eri näkymien mallintaminen ja tietojen esittämistapojen valinta (Kuva 4.). Tietokantahaku oli aina tarkoitus optimoida SQL-hakukielellä, jotta haku kohdistuu vain niihin tietoihin, joita on tarkoitus hakea. Visual C# ja JavaScript -kielillä toteutettiin WWW-sovelluksen toiminnallisuudet. CSS-kielillä toteutettiin tyyliohjeisiin liittyvät seikat, jolloin saman tyylin tiedot haettiin aina erillisestä tyyliohjetiedostosta. XHTML-merkintäkielellä toteutettiin WWW-sovelluksen graafinen puoli, johon kuului esimerkiksi kaavioiden piirtäminen tietokannasta haettujen tietojen perusteella. WWW-sovelluksen kehityksen loppuksi, sovelluksella suoritettiin ennalta määriteltäviä testitapauksia, jotka osoittivat WWW-sovelluksen olevan toimiva ja käytettävä.

Right leg

Maximum amount of motions:

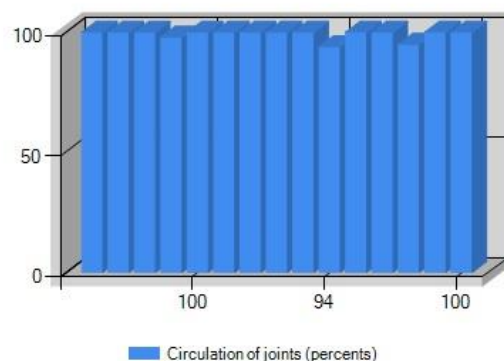
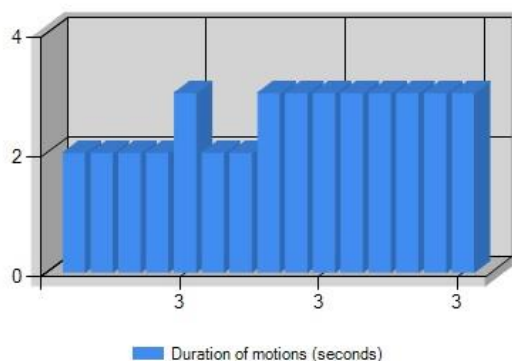
15

Correctly done motions:

12

Correctly done motions by percents:

80%



Kuva 4. Näkymä jalan kuntoutustiedoista WWW-sovelluksessa.

5 YHTEENVETO

Tässä opinnäytetyössä tutkittiin terveystietojen tallentamista sähköisiin tietokantoihin. Tietokantojen käyttämistä tarkasteltiin terveysteknologian näkökulmasta. Tietokantahakuja tarkasteltiin sekä terveydenhuollon ammattilaisten että asiakkaiden näkökulmista. Opinnäytetyö jaettiin kahdeksi osaksi, jotka olivat teoriaosuus ja projektityö. Teoriaosuuden pohjatiedoiksi käsiteltiin terveysteknologiassa käytettäviä sähköisiä terveystietokantoja, joiden tueksi esiteltiin suomalaisia kansallisia terveystietokantapalveluja, joista voidaan katsoa omia terveystietoja Internet-yhteyden avulla. Projektityöosuudessa kerrottiin ensin projektityössä käytetyistä kehitysvaiheista, kehitystyökaluista, kehitysmenetelmistä ja kielistä. Tämän jälkeen avattiin, miten käytännössä luodaan Microsoft Visual Studiolla WWW-sovellus, jolla voidaan tehdä tietokantahakuja luotuun tietokantaan ja esittää haettuja tietoja halutussa muodossa.

Tietokantojen käyttäminen terveysteknologiassa edellyttää vahvaa tuntemusta terveydenhuoltoalasta, vallitsevasta lainsäädännöstä ja tietoturvasta. Terveydenhuoltoalalla ollaan tekemisissä ihmisten terveyden kanssa, mitä on haluttu korostaa tiukan lainsäädännön avulla. Esimerkiksi potilastietojen katseleminen on sallittu vain potilaan hoitamiseen osallistuvilta henkilöiltä ja tietojen tarkasteleminen täytyy aina liittyä hoidon edistämiseen. Kriittisistä terveystiedoista täytyy olla tarvittavat varmuuskopiot, sillä tietokantapalvelimet eivät kestä ikuisesti.

Esiteltyjä kansallisia terveystietokantapalveluja olivat Omakanta ja Taltioni. Omakanta on maksuton palvelu, jonka avulla omia terveystietoja voidaan tarkastella paikasta ja ajasta riippumatta Internet-yhteyden välityksellä. Palvelun käyttöehdot hyväksymällä pääsee tarkastelemaan esimerkiksi omista lääkärikäynneistä tallennettuja tietoja ja määrättyjä sähköisiä reseptejä. Palveluun tunnustaudutaan verkkopankkitunnuksilla, sähköisellä henkilökortilla tai mobiilivarmenteella. Toinen tarkasteltu palvelu oli Taltioni, joka on sähköisessä muodossa oleva terveysarkisto. Sinne voidaan myös tallentaa käyttäjän tärkeäksi kokemia hyvinvointiin liittyviä asioita perusterveydenhuollon lisäksi. Palvelun käyttö aloitetaan luomalla oma Taltioni-terveystili ja palveluun tunnustaudutaan verk-

kopankkitunnuksilla tai matkapuhelimen SIM-kortilla olevalla mobiilivarmenteella.

Projektityöosuudessa oli tarkoituksena kehittää toimeksiantajalle prototyyppi tietokantaa hyödyntävästä WWW-sovelluksesta, jolla voidaan seurata kuntoutettavan ihmisen kehitystä. Tietojen esittäminen toteutettiin sekä tekstinä että graafisina kaavioina. Suurempien tietomäärien esittämisessä käytettiin kaavioita, sillä niiden avulla tiedot saatiin mahtumaan pienempään ja havainnollistavampaan tilaan. Projektityöosuuden koko oli opinnäytetyössä pienempi, koska suurin osa opinnäytetyöhön käytettävästä ajasta kului WWW-sovelluksen kehittämisen opetteluun, sillä WWW-sovelluskehityksestä ei ollut paljon aiempaa kokemusta. Projektin lopputuotteesta tuli määritetyssä aikataulussa vaatimukset täyttävä eli laadukas. Tehty prototyyppi toimii toimeksiantajalle mallina suurempaa kokonaisuutta varten. Opinnäytetyön raportti toimii dokumentaationa lopullisen järjestelmän tuotekehityksessä.

LÄHTEET

- [1] Tourela, H. 2010. Potilaiden mahdollisuudet käyttää uusia terveysteknologiaa sisältäviä palveluja? Viitattu 5.1.2015 <http://kotu.oulu.fi/wellbe4/esitykset/tuorila.pdf>.
- [2] Soininen, M. 2010. Terveysteknologia tulee arkeen. Viitattu 5.1.2015 http://www.laakarilehti.fi/uutinen.html?opcode=show/news_id=9212/type=1.
- [3] Laki terveydenhuollon laitteista ja tarvikkeista 24.6.2010/629.
- [4] Sosiaali- ja terveysalan lupa- ja valvontavirasto 2010. Terveysteknologiaa sisältävien laitteiden ja tarvikkeiden valmistajan vaaratilanneilmoitukset. Viitattu 5.1.2015 http://www.valvira.fi/files/tiedostot/m/a/maarays_1_2010_valmistajan_vt_ilmointus.pdf.
- [5] Keppola, E. 2014. Sote-uudistus voi yhtenäistää potilastietojärjestelmiä. Viitattu 5.1.2015 <http://www.taloussanomat.fi/terveydenhuolto/2014/07/29/sote-uudistus-voi-yhtenaistaa-potilastietojarjestelmia/20149605/12>.
- [6] Väestörekisterikeskus 2013. Henkilötunnus. Viitattu 6.1.2015 <http://www.vrk.fi/default.aspx?id=167>.
- [7] Pullen, E. 2010. Why there needs to be a national healthcare information database. Viitattu 6.1.2015 <http://www.kevinmd.com/blog/2010/01/national-healthcare-information-database.html>.
- [8] Cardon, E. 2011. Understanding The Healthcare Database: Purposes, Strengths, and Weaknesses. Viitattu 6.1.2015 <https://www.healthcatalyst.com/healthcare-database-purposes-strengths-weaknesses>.
- [9] Martin, G. 2008. The essential nature of healthcare databases in critical care medicine. Viitattu 6.1.2015 <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2592733/>.
- [10] Rikoslaki 19.12.1889/39.
- [11] Henkilötietolaki 22.4.1999/523.
- [12] Laki potilaan asemasta ja oikeuksista 17.8.1992/785.
- [13] Omakanta 2014. Omakanta. Viitattu 10.1.2015 <http://www.kanta.fi/omakanta>.
- [14] Taltioni 2014. Taltioni. Viitattu 17.1.2015 <http://taltioni.fi/>.
- [15] Vaasan yliopisto 2010. Tietokanta. Viitattu 18.1.2015 http://lipas.uwasa.fi/~jt/tte_103/tietokannat.ppt.
- [16] Sarja, J. 2006. Mikä on tietokanta? Viitattu 18.1.2015 <http://verkkopedagogi.net/vanhat/fi/sisalto/materiaalit/access2003/luku021c5a.html>.
- [17] Oulun seudun ammattiopisto 2013. 3. Tietokantojen peruskäsitteet. Viitattu 18.1.2015 http://www.okol.org/verkkokurssit/datanomi/tietojarjestelmien_kehittaminen/tiedonhallintajarjestemat/peruskasitteet.htm.
- [18] Nations, D. 2011. What is a Web Application? Viitattu 23.1.2015 http://webtrends.about.com/od/webapplications/a/web_application.htm.
- [19] Microsoft 2010. Introducing Visual Studio. Viitattu 25.1.2015 [https://msdn.microsoft.com/en-us/library/fx6bk1f4\(v=vs.100\).aspx](https://msdn.microsoft.com/en-us/library/fx6bk1f4(v=vs.100).aspx).

- [20] Microsoft 2014. Microsoft Visual Studio and .NET Framework Documentation (ISO image). Viitattu 25.1.2015 <http://www.microsoft.com/en-us/download/details.aspx?id=34794>.
- [21] Goodwill Community Foundation 2015. What is an application? Viitattu 25.1.2015 <http://www.gcflearnfree.org/computerbasics/3>.
- [22] Microsoft 2012. Introduction to Windows Service Applications. Viitattu 25.1.2015 [https://msdn.microsoft.com/en-us/library/d56de412\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/d56de412(v=vs.110).aspx).
- [23] W3 Group Finland Oy 2014. Kolme tapaa kehittää mobiilisovellus. Viitattu 30.1.2015 <http://w3.fi/kolme-tapaa-kehittaa-mobiilisovellus/>.
- [24] Microsoft 2014. Microsoft SQL Server. Viitattu 30.1.2015 <https://msdn.microsoft.com/en-us/library/bb545450.aspx>.
- [25] Microsoft 2014. Use SQL Server Management Studio. Viitattu 30.1.2015 [https://msdn.microsoft.com/en-us/library/ms174173\(v=sql.120\).aspx](https://msdn.microsoft.com/en-us/library/ms174173(v=sql.120).aspx).
- [26] Microsoft 2013. Yhteyden muodostaminen SQL Server -tietokantaan. Viitattu 30.1.2015 <https://support.office.com/fi-fi/article/Yhteyden-muodostaminen-SQL-Server--tietokantaan-e5004318-0f2e-46a3-8b15-1559aa3c04db>.
- [27] Microsoft 2010. .NET Framework 4. Viitattu 7.2.2015 [https://msdn.microsoft.com/en-us/library/w0x726c2\(v=vs.100\).aspx](https://msdn.microsoft.com/en-us/library/w0x726c2(v=vs.100).aspx).
- [28] Microsoft 2010. .NET Framework Conceptual Overview. Viitattu 7.2.2015 [https://msdn.microsoft.com/en-us/library/zw4w595w\(v=vs.100\).aspx](https://msdn.microsoft.com/en-us/library/zw4w595w(v=vs.100).aspx).
- [29] Microsoft 2010. ASP.NET Overview. Viitattu 14.2.2015 [https://msdn.microsoft.com/en-us/library/4w3ex9c2\(v=vs.100\).aspx](https://msdn.microsoft.com/en-us/library/4w3ex9c2(v=vs.100).aspx).
- [30] Microsoft 2014. ASP.NET Overview. Viitattu 14.2.2015 [https://msdn.microsoft.com/en-us/library/4w3ex9c2\(v=vs.140\).aspx](https://msdn.microsoft.com/en-us/library/4w3ex9c2(v=vs.140).aspx).
- [31] Microsoft 2013. Introduction to the C# Language and the .NET Framework. Viitattu 23.2.2015 <https://msdn.microsoft.com/en-us/library/z1zx9t92.aspx>.
- [32] File-Extension.org 2015. File formats starting with a letter C. Viitattu 23.2.2015 <http://www.file-extension.org/alphabetical/c>.
- [33] The Web Standards Project 2003. HTML Versus XHTML. Viitattu 9.3.2015 <http://www.webstandards.org/learn/articles/askw3c/oct2003/>.
- [34] Mozilla Developer Network 2014. Information: Why use CSS? Viitattu 9.3.2015 https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Getting_started/Why_use_CSS.
- [35] HTML.net 2012. Lesson 1: What is JavaScript? Viitattu 10.3.2015 <http://html.net/tutorials/javascript/lesson1.php>.
- [36] Mozilla Developer Network 2014. What is JavaScript? Viitattu 10.3.2015 https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript.
- [37] NTC Hosting 2014. SQL (Structured Query Language). Viitattu 14.3.2015 <http://www.ntchosting.com/encyclopedia/databases/structured-query-language/>.